REMARKS

I. Status of the Application

This amendment is filed in response to the office action dated April 15, 2009. Claims 1-11, 13-18, 20-26, and 28-33 were pending and were rejected. By way of this response, claims 1, 7, 8, 11, 15-18, 22-24, and 26 are amended and claims 5 and 6 are canceled. Thus, claims 1-4, 7-11, 13-17, 20-26, and 28-33 are pending and at issue.

II. Rejections under 35 U.S.C. §103

Claims 1-11, 13, 14, 18, 20, 21, 25, 26, and 28-33 were rejected under 35 USC §103(a) as being unpatentable over Chheda (US Pub. No. 20050114850) in view of Chen (Energy-Aware Compilation and Execution in Java-Enabled Mobile Devices, 2003 IEEE, pp. 1-8). The applicants respectfully traverse this rejection.

"To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art." MPEP §2143.03. As will be explained further with reference to specific claims, the office action failed to establish that the alleged combination of Chheda and Chen teaches, discloses, or suggests each and every element of each of claims 1-11, 13, 14, 18, 20, 21, 25, 26, and 28-33.

Independent claims 1, 11, 18, and 26 generally recite power and/or energy optimized compiling and/or execution of code from received non-native instructions. Each of claims 1, 11, 18, and 26 generally recite:

> determining an initial number of times to interpretively execute the plurality of non-native instructions;
> interpretively executing the plurality of non-native instructions the initial number of times; and
> monitoring execution of the plurality of non-native instructions to determine when the plurality of non-native instructions have been interpretively executed the initial number of times;
> compiling the plurality of non-native instructions to generate object code for the non-native instructions only after interpretively executing the plurality of non-native instructions the initial number of times…

At least this combination of elements is not disclosed or suggested by the alleged combination of Chheda and Chen.

Previous claims 5 and 6 included most of the above limitations that, by amendment, are now generally recited in each of independent claims 1, 11, 18, and 26. Differences in syntax and dependencies and the addition of the "monitoring..." step (as described at paragraph [0034] of the original specification) further clarify those previous recitations in the context of the independent claims. The action admits Chheda does not disclose "determining..." in the rejection of claim 6 (pages 6 and 7 of the office action), and "interpretively executing..." and "compiling... only after..." in the rejection of claim 5 (page 6 of the office action) and cites Chen for disclosing the same at Sec. 1, para. 5 and Sec. 3, para. 2. Chen, however, generally describes receiving non-native code, identifying (or determining) which code segments are able to be remotely compiled and executed, and dynamically deciding which remote resource is best able to compile and execute the identified segment(s). In particular, Chen generally describes that some JVMs use interpretive execution (Sec. 1, para. 5). Also, energy consumption profiles for a device that receives non-native code are created by tracking the energy consumption of the device processor and memory system and that the energy cost of the received code is estimated by counting the number of instructions of each type within the code as multiplied by the base energy consumption of the corresponding instruction (Sec. 2, para. 3). In one embodiment described by Chen, the code segments that are able to be remotely executed are able to be identified by "automatic tools that make use of profile information" (Sec. 3, para. 2). At each invocation of the identified code, the JVM disclosed by Chen dynamically decides whether to execute it locally or remotely.

Simply determining which received code segments are able to be compiled and executed remotely and the mere possibility of interpretive execution as described by Chen cannot teach or suggest the recitations of claims 1, 11, 18, and 26, as generally described above. To contrast the description of Chen and the claim recitations, Chen determines which received code segments are able to be remotely compiled and executed, while claims 1, 11, 18, and 26, generally recite determining how many times to interpretively execute the received code. Chen describes

tracking the energy consumption of the device to create an energy profile of the processor and memory system, while claims 1, 11, 18, and 26 generally recite monitoring execution to determine when the plurality of non-native instructions have been interpretively executed the initial number of times. Further, Chen generally describes that some JVMs are capable of interpretive execution and dynamically determining whether to remotely compile and execute the identified code, while claims 1, 11, 18, and 26 generally recite compiling the received code only after it has been interpretively executed the initial number of times. As admitted in the rejection of claims 5 and 6 discussed on pages 6 and 7 of the office action, Chedda does not teach or suggest these elements, either. At least for these reasons, the alleged combination of Chedda and Chen does not render claims 1, 11, 18, and 26 unpatentable.

With regard to claims 2-4, 7-10, 13, 14, 20, 21, and 28-33, which depend from claims 1, 11, 18, and 26, respectively, the applicants respectfully submit that the alleged combination of Chheda and Chen does not render claims these claims unpatentable at least for the same reasons as claims 1, 11, 18, and 26.

III. Rejections under 35 U.S.C. §102(e)

Claims 15-17 and 22-24 were rejected under 35 USC §102(e) as being anticipated by Chedda. The applicants respectfully traverse this rejection.

Independent claims 15 and 22 generally recite power and/or energy optimized compiling and/or execution of code from received non-native instructions. Each of claims 15 and 22 generally recite:

> determine an initial number of times to interpretively execute a plurality of non-native instructions;
> interpretively execute the plurality of non-native instructions the initial number of times;
> monitor execution of the non-native instructions to determine when the non-native instructions have been interpretively executed the initial number of times;
> compile the non-native instructions to generate object code for the non-native instructions only after interpretively executing the plurality of non-native instructions the initial number of times…

As discussed above, previous claims 5 and 6 generally recited the above elements that, by amendment, are now recited in each of independent claims 15 and 22. In the §103(a) rejection, the office action admits on pages 6 and 7 that Chedda does not disclose the recitations of previous claims 5 and 6 and cites Chen for disclosing the same. As discussed above regarding the §103(a) rejection of claims 1-11, 13, 14, 18, 20, 21, 25, 26, and 28-33, Chen does not describe these recitations. Therefore, at least this combination of elements is not disclosed or suggested by Chheda and Chen. At least for these reasons, neither Chedda nor Chen anticipates claims 15 and 22.

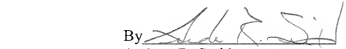
With regard to claims 16, 17, and 23-25, which depend from claims 15 and 22, respectively, the applicants respectfully submit that neither Chheda nor Chen anticipates these claims at least for the same reasons as claims 15 and 22.

IV. Conclusion

In view of the above, Applicants submit that the pending application is in condition for allowance and an early action so indicating is respectfully requested.

July 13, 2009

Respectfully submitted,

By 

Andrew R. Smith
Registration No.: 62,162
MARSHALL, GERSTEIN & BORUN LLP
233 S. Wacker Drive, Suite 6300
Sears Tower
Chicago, Illinois 60606-6357
(312) 474-6300
Attorney for Applicant